DOCUMENT RESUME

ED 217 871 IR 010 257

AUTHOR Mapou, Robert L.
TITLE The Microcomputer as Tachistoscope: Trials and Tribulations of Timing on the TRS-80.
PUB DATE Mar 82
NOTE 14p.; Paper presented at the Annual Meeting of the Southeastern Psychological Association (28th, New Orleans, LA, March 24-27, 1982).

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Programs; Computer Science; Input Output Devices; Measurement Equipment; *Microcomputers; *Programing; Programing Languages; Science Equipment; *Tachistoscopes
IDENTIFIERS Assembler Languages; BASIC Programing Language; Reaction Timing; Stimulus Timing; *TRS 80

ABSTRACT

In this paper a hardware/software system is described which permits the use of the Radio Shack TRS-80 Model I as a tachistoscope in perceptual-cognitive research. Hardware requirements and configuration, problems in developing software routines to handle reaction and stimulus timing using the TRS-80, and the development of a routine for tachistoscopic stimulus display using the TRS-80 are discussed. The limitations of the TRS-80 as a tachistoscope are outlined and ways to overcome those limitations are presented. Accompanying the text are an 8-item reference list, a block diagram of the hardware configuration for using the TRS-80 as a tachistoscope, a figure illustrating the assembly language tachistoscopic display and reaction time routine, and a figure illustrating portions of the BASIC program for tachistoscopic display. (Author/JL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED217871

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

The Microcomputer as Tachistoscope: Trials and Tribulations of Timing on the TRS-80

Robert L. Mapou

Department of Psychology

Emory University

Atlanta, Georgia 30322

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Robert L. Mapou

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

NOT FOR CITATION WITHOUT PERMISSION OF AUTHOR

Paper presented at the 28th Annual Meeting of the Southeastern Psychological Association, New Orleans, Louisiana, March 24-27, 1982

IR 010 257

Abstract

A hardware/software system is described which permits use of the TRS-80 Model I microcomputer as a tachistoscope. Limitations of the TRS-80 as a tachistoscope are discussed and ways to overcome the limitations are provided. Hardware and software configurations for reaction timing, stimulus timing, and stimulus presentation are presented and discussed.

The Radio Shack TRS-80 Model I is a popular microcomputer which has experienced increasing use in perceptual-cognitive laboratories. The TRS-80 is a Z-80 based microcomputer with many sophisticated features. It can be connected to a range of peripheral devices, including disk drives for data storage, external clocks for accurate timing, and subject response panels. Unfortunately, the TRS-80 alone is less than adequate for the requirements of perceptual-cognitive research. Knowledge of the limitations of the machine, as well as both the hardware and software required to overcome these limitations, are a 'must' for the researcher. This paper addresses three problem areas and their solutions associated with application of the TRS-80 as a tachistoscope. These aspects are hardware requirements and configuration, reaction and stimulus timing, and stimulus presentation.

Hardware Requirements and Configuration

The TRS-80 alone is unable to provide timing accurate to the millisecond. An accurate timing device must be provided externally to the computer. One possible configuration using the LVB Corporation interface has been described by Grice (1981).

The system employed in this laboratory is illustrated in Figure 1. It consists of a TRS-80 Model I, Expansion Interface, two disk drives, a serial printer, and several other peripheral devices, including an external clock and subject response panel. An HUH Mini 8100 (California Computer Systems, 308 Laurelwood Road, Santa Clara, California 95050) is connected to the TRS-80 bus extension card edge on the expansion interface. The Mini 8100 is a peripheral interface which allows S-100 bus compatible devices to be connected to the TRS-80. This approach increases the range of peripheral devices available to the TRS-80 and allows devices to be accessed from software routines. The Mini 8100 can hold up to four S-100 compatible cards. A

100,000 day clock (Mountain Hardware, Inc., 300 Harvey West Blvd., Santa Cruz, California 95060) is plugged into one of the sockets on the Mini 8100. The clock runs continuously and records time to 100 microsecond accuracy. Each of fifteen addressable ports on the clock provides a single digit (0 to 6 or 9) which indicates the current time in the units denoted by that particular port. Port one provides the time in units of 100 milliseconds. Each subsequent port provides the time in units which are ten times that of the preceding port, up to units of 10,000 days. Computing the time at a given point involves reading the values of the required ports and performing a sum-of-products operation.[1] To obtain a reaction time, one would read the clock at the start of the reaction time interval, read the clock following subject response, compute the two times, and subtract the first value from the second value.

Reaction and Stimulus Timing

Most programming on the TRS-80 Model I is done in BASIC, a relatively simple and easily learned language. Several early attempts were made to do reaction timing and stimulus timing in BASIC, both in this laboratory and elsewhere (Perera, 1978, 1979). However, it became clear that BASIC could not perform timing with the type of accuracy required, even when using an external clock. This is due to the fact that BASIC is an interpreted language: each program line is individually read by the computer, translated into machine language, and executed. This is time-consuming and limits the accuracy of any timing method. Thus, the decision was made to do all timing in assembly language.

Reaction timing is straightforward: clock readings are stored at the beginning of the interval and following the subject's response. To compute the reaction time, it is necessary to perform the calculations noted above.

Because no multiply operation was readily available in assembly language on the TRS-80, it was decided to record and store the clock readings via the assembly language routine and compute the reaction time in the main BASIC program. Lines 230 to 340 in the assembly language routine in Figure 2 store the readings at the start of the interval. Lines 610 to 720 in the routine store the readings following the subject's response. The BASIC program lines in Figure 3(A) call the assembly language routine, obtain the values stored by the routine, and compute the reaction time in milliseconds.

Stimulus timing proved to be more complex. A further complication is the inaccuracy in actual stimulus display time due to video screen scanning rates. This problem has not yet been addressed in this laboratory, although several solutions have been described (Grice, 1981; Lincoln and Lane, 1980; Reed, 1979). The focus in this laboratory has been on timing of the interval between sending the stimulus to the video screen and removing it from the screen.

Stimulus timing with a free-running clock requires the following steps: 1) place the stimulus on the screen and read the time at that point; 2) continuously monitor the time on the clock; 3) remove the stimulus from the screen when the difference between the current time and the initial time is the desired value. As noted, computation of the time at any point requires a sum-of-products operation. Again, because of the problem with multiplication in assembly language at the time this system was developed, and because stimulus timing in BASIC would have taken more time than the stimulus presentation time itself, it was decided to forego stimulus timing via the clock and turn to software methods of timing.

The method chosen involves displaying the stimulus, entering a loop to provide the desired stimulus display time, and exiting the loop and clearing the screen when the time has passed. The method is similar to that described

by Owings and Fiedler (1979). Lines 380 to 450 in Figure 2 comprise loop initialization and the timing loop. The number of loops required for the desired times (60 to 160 ms, in units of 20 ms) was obtained using a separate program and the external clock. The number of loops is passed as a parameter to the assembly routine.

Stimulus Display

Positions on the screen of the TRS-80 occupy a segment of memory and can be addressed directly from software. Thus, stimuli may be displayed by sending the appropriate graphics codes to the appropriate memory locations on the screen. Initial programs developed in the laboratory displayed stimuli by sending graphics characters to the screen via the POKE command in BASIC. However, the slow execution speed of BASIC prevented stimuli from appearing on the screen at one time. Subjects could see the stimuli being "drawn" on the screen and did not perceive the desired tachistoscopic effect. Thus, another approach had to be taken.

Unlike the APPLE II microcomputer, which allows storage of stimuli in a select area of memory (these are called "display pages", see Cavanagh and Anstis, 1980), the TRS-80 provides no such display areas. It was decided to set aside a specific portion of memory to simulate a display page for stimulus storage. The graphics codes could then be sent to these memory locations via the POKE commands in BASIC. Once present in memory, an assembly language routine could transfer the contents of the pseudo-display page to the screen at a rate which would make the stimulus appear on the screen simultaneously (ignoring scanning rate delays), producing a tachistoscopic effect. The BASIC program lines in Figure 3(B) send the graphics codes to the "display page" in memory to create the stimuli; in this case, two patterns of four small squares. Lines 190 to 220 in Figure 2 send the "display page" to the screen.

This report is brief and lacks much of the detail necessary to implement this type of system in its entirety. A more detailed report is being prepared for publication. It is clear, however, that implementation of this type of system requires complex programming and knowledge of the memory structure of the TRS-80 or other microcomputers. It is suggested that all laboratories planning to use microcomputers in this fashion need to consider having at least one person present who is knowledgeable of assembly language programming and computer architecture in order to insure that the type of accuracy necessary to do this type of research can be developed. Additional information on similar system development can be found in the references.

Footnote

[1]For example, suppose interval timing up to 9.999 seconds in millisecond accuracy is desired. At a given point, one would need to read the seconds, hundreds of milliseconds, tens of milliseconds, and millisecond ports. Assuming the values read from these ports are, respectively, 2, 1, 5, and 3, the time is obtained by computing 2(1000) + 1(100) + 5(10) + 3 = 2153 milliseconds.

References

Cavanagh, P. and Anstis, S. M. Visual psychophysics on the Apple II: getting started. Behavior Research Methods & Instrumentation, 1980, 12, 614-626.

Grice, G. R. Accurate reaction time research with the TRS-80 microcomputer. Behavior Research Methods & Instrumentation, 1981, 13, 674-676.

Lincoln, C. E. and Lane, D. M. Reaction time measurement errors resulting from the use of CRT displays. Behavior Research Methods & Instrumentation, 1980, 12, 55-57.

Owings, R. A. and Fiedler, C. H. Measuring reaction time with millisecond accuracy using the TRS-80 microcomputer. Behavior Research Methods & Instrumentation, 1979, 11, 589-591.

Perera, T. B. A versatile microcomputer-based multiple-field tachistoscope. Behavior Research Methods & Instrumentation, 1978, 10, 546-547.

Perera, T. B. A comparison of BASIC language timing loops for the TRS-80 microcomputer. Behavior Research Methods & Instrumentation, 1979, 11, 592.

Reed, A. V. Microcomputer display timing: problems and solutions. Behavior Research Methods & Instrumentation, 1979, 11, 572-576.

Figure Legends

Figure 1. Block diagram of the hardware configuration in this perceptual-cognitive laboratory.

Figure 2. Assembly language tachistoscopic display and reaction time routine.

Figure 3. Portions of main BASIC program: (A) lines which call the tachistoscopic assembly language routine and compute the reaction time; (B) lines which send the graphics codes comprising the stimuli to the "display page" in memory.

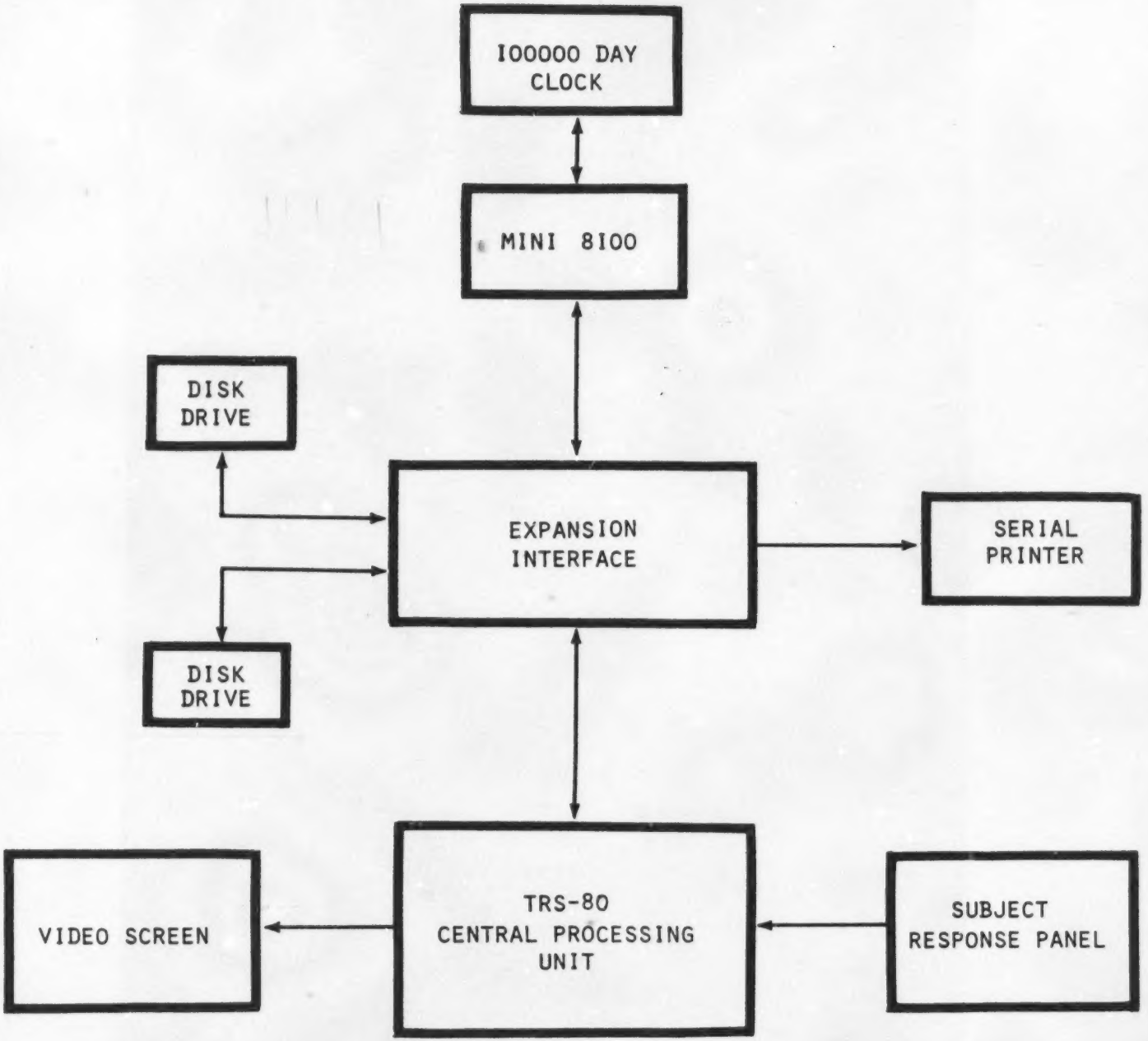
100000 DAY CLOCK
MINI 8100
DISK DRIVE
EXPANSION INTERFACE
SERIAL PRINTER
DISK DRIVE
VIDEO SCREEN
TRS-80 CENTRAL PROCESSING UNIT
SUBJECT RESPONSE PANEL

FIGURE 1

```
F650            00010          ORG     0F650H          ;62456 DECIMAL
FBFF            00020 VSCREN   EQU     0FBFFH          ;VIRTUAL SCREEN ADR
F7FF            00030 BSCREN   EQU     0F7FFH          ;BLANK SCREEN ADR
3C00            00040 SCREEN   EQU     3C00H           ;SCREEN ADR
3810            00050 KEYBD    EQU     3810H           ;KEYBD BUFFER ADR
0400            00060 COUNT    EQU     400H            ;XFER 1024 BYTES
F650 F3         00070          DI                      ;DISABLE INTERRUPTS
F651 F5         00080          PUSH    AF              ;SAVE ALL REGISTERS
F652 D5         00090          PUSH    DE
F653 C5         00100          PUSH    BC
F654 DDE5       00110          PUSH    IX
F656 FDE5       00120          PUSH    IY
F658 CD7F0A     00130 INIT     CALL    0A7FH           ;GET DELAY TIME
F65B E5         00140          PUSH    HL              ;STORE FOR LATER
F65C 11003C     00150          LD      DE,SCREEN       ;SCREEN START ADR
F65F 21FFF7     00160          LD      HL,BSCREN       ;BLANK SCRN START ADR
F662 010004     00170          LD      BC,COUNT        ;NO OF BYTES
F665 EDB0       00180          LDIR                    ;CLEAR SCREEN
F667 11003C     00190          LD      DE,SCREEN       ;SCREEN STARTING ADR
F66A 21FFFB     00200          LD      HL,VSCREN       ;VIR SCRN STARTING ADR
F66D 010004     00210          LD      BC,COUNT        ;NO BYTES TO XFER
F670 EDB0       00220          LDIR                    ;XFER STIMULUS TO SCREEN
F672 21DCF6     00230 TIME1    LD      HL,CLK1         ;START TIME STORAGE ADR
F675 0625       00240          LD      B,37            ;10X SEC PORT FROM CLOCK
F677 48         00250          LD      C,B             ;TRANSFER TO C FOR ADR
F678 EDA2       00260          INI                     ;INPUT 10X SEC AND STORE
F67A 48         00270          LD      C,B             ;UPDATE PORT ADR
F67B EDA2       00280          INI                     ;INPUT SECONDS AND STORE
F67D 48         00290          LD      C,B             ;UPDATE PORT ADR
F67E EDA2       00300          INI                     ;INPUT 1/10 SEC AND STORE
F680 48         00310          LD      C,B             ;UPDATE PORT ADR
F681 EDA2       00320          INI                     ;INPUT 1/100 SEC & STORE
F683 48         00330          LD      C,B             ;UPDATE PORT ADR
F684 EDA2       00340          INI                     ;INPUT MS AND STORE
                00350 ,
                00360 ,
F686 E1         00370          POP     HL              ;RETRIEVE DELAY TIME
F687 010100     00380          LD      BC,0001H        ;INITIALIZE REGISTERS
F68A DD21BEFF   00390          LD      IX,0FFBEH       ;65 COUNTS TILL CARRY
F68E FD211038   00400          LD      IY,KEYBD        ;KEYBOARD SCAN LOCATION
F692 DD09       00410 LOOP     ADD     IX,BC           ;INCR LOOP COUNTER
F694 30FC       00420          JR      NC,LOOP         ;JP NOT YET 1 MS
F696 DD21BEFF   00430          LD      IX,0FFBEH       ;RESET LOOP COUNTER
F69A ED42       00440 TEST     SBC     HL,BC           ;1 MS HAS PASSED
F69C 20F4       00450          JR      NZ,LOOP         ;JP TIMING NOT DONE
F69E 11003C     00460          LD      DE,SCREEN       ;SCREEN START ADR
F6A1 21FFF7     00470          LD      HL,BSCREN       ;BLANK SCRN START ADR
F6A4 010004     00480          LD      BC,COUNT        ;NO BYTES TO XFER
F6A7 EDB0       00490          LDIR                    ;CLEAR SCREEN
F6A9 21E6F6     00500          LD      HL,CHAR         ;LOC OF CHAR NOT ALLOWED
F6AC 56         00510          LD      D,(HL)          ;XFER CHAR TO D
F6AD 78         00520          LD      A,B             ;CLEAR REG A
F6AE FD8600     00530 LOOP2    ADD     A,(IY)          ;CHECK KEYBD
F6B1 28FB       00540          JR      Z,LOOP2         ;JP NO INPUT
F6B3 BA         00550          CP      D               ;CHECK INPUT CHAR
F6B4 2004       00560          JR      NZ,TIME2        ;JP IF ALLOWED
F6B6 78         00570          LD      A,B             ;CLEAR A AGAIN
F6B7 C3AEF6     00580          JP      LOOP2           ;CONTINUE
                00590 ,
                00600 ,
F6BA 21E1F6     00610 TIME2    LD      HL,CLK2         ;FIN TIME STORAGE ADR
F6BD 0625       00620          LD      B,37            ;10X SEC PORT FROM CLOCK
F6BF 48         00630          LD      C,B             ;TRANSFER TO C FOR ADR
F6C0 EDA2       00640          INI                     ;INPUT 10X SEC AND STORE
F6C2 48         00650          LD      C,B             ;UPDATE PORT ADR
F6C3 EDA2       00660          INI                     ;INPUT SECONDS AND STORE
F6C5 48         00670          LD      C,B             ;UPDATE PORT ADDRESS
F6C6 EDA2       00680          INI                     ;INPUT 1/10 SEC AND STORE
F6C8 48         00690          LD      C,B             ;UPDATE PORT ADDRESS
F6C9 EDA2       00700          INI                     ;INPUT 1/100 SEC & STORE
F6CB 48         00710          LD      C,B             ;UPDATE PORT ADR
F6CC EDA2       00720          INI                     ;INPUT MS AND STORE
                00730 ,
                00740 ,
F6CE 6F         00750 BACK     LD      L,A             ;SAVE CHARACTER
F6CF 2600       00760          LD      H,00            ;ZERO REGISTER H
F6D1 FDE1       00770          POP     IY              ;RESTORE REGISTERS
F6D3 DDE1       00780          POP     IX
F6D5 C1         00790          POP     BC
F6D6 D1         00800          POP     DE
F6D7 F1         00810          POP     AF
F6D8 FB         00820          EI                      ;ENABLE INTERRUPTS
F6D9 C39A0A     00830          JP      0A9AH           ;EXIT TO BASIC
                00840 ,
                00850 ,
F6DC 00         00860 CLK1     DEFB    0               ;START TIMES - 10X SEC
F6DD 00         00870          DEFB    0               ; SEC
F6DE 00         00880          DEFB    0               ; 1/10 SEC
F6DF 00         00890          DEFB    0               ; 1/100 SEC
F6E0 00         00900          DEFB    0               ; 1/1000 SEC
F6E1 00         00910 CLK2     DEFB    0               ;END TIMES - 10X SEC
F6E2 00         00920          DEFB    0               ; SEC
F6E3 00         00930          DEFB    0               ; 1/10 SEC
F6E4 00         00940          DEFB    0               ; 1/100 SEC
F6E5 00         00950          DEFB    0               ; 1/1000 SEC
F6E6 00         00960 CHAR     DEFB    0               ;CHAR CODE NOT ALLOWED
0000            00970          END
00000 TOTAL ERRORS
32293  TEXT AREA BYTES LEFT
```


FIGURE 2

(A)

```
1530  REM Prompt the subject for the next trial
1540  PRINT@448,CHR$(23);"WHEN READY, PRESS THE FOOT PEDAL"
1550  PRINT@521,CHR$(23);"AND LOOK AT THE DOT"
1555  '
1560  REM Wait for the subject to start the trial
1570  IF INKEY$="3" THEN CLS ELSE1570
1575  '
1580  REM Send the stimulus to the screen and wait for a response
1590  REM Clear the screen following the response
1600  IX=USRO(S1TIME):CLS
1605  '
1610  REM Get the clock readings from where they were stored
1620  FOR M=1 TO 10
1630  T(M)=PEEK(RAM%+M)
1640  T(M)=T(M)-INT(T(M)/16)*16
1650  NEXT M
1655  '
1660  REM Compute the reaction time in milliseconds
1670  TIME=T(10)+10*(T(9)+10*(T(8)+10*(T(7)+10*T(6))))
1680  TIME=TIME-(T(5)+10*(T(4)+10*(T(3)+10*(T(2)+10*T(1)))))
```

(B)

```
1250  REM Send stimuli A1 and A2 to the display page
1255  REM O1 and O2 are used to change the positions of the stimuli
1260  POKE 15838+O1,179:POKE 15840+O1,179: REM A12
1270  POKE 15838+O2,140:POKE 15839+O2,179:POKE 15840+O2,140
1280  GOTO 1530
```

FIGURE 3